MULTIMEDIA UNIVERSITY

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2015/2016

TSW7021 – DATA STRUCTURES AND OBJECT-ORIENTED PROGRAMMING

(All sections / Groups)

18 JANUARY 2016
8.00 p.m - 10.00 p.m
(2 Hours)

INSTRUCTIONS TO STUDENTS

1. This Question paper consists of 5 pages with 4 Questions only.

2. Attempt **ALL** questions. Each question carries equal marks and the distribution of the marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

Question 1 (10 marks)

```
public class Bicycle {
    protected int gear;
    protected int speed;

    public void setGear(int newValue) {
        gear = newValue;;
    }

    public void applyBrake(int decrement) {
        speed -= decrement;
    }

    public void speedup(int increment) {
        speed += increment;
        System.out.println("Speeding");
    }
}

public class SpecialBike extends Bicycle{


    public void applyBrake(int decrement) {
        speed -= (decrement*1.5);
    }

    public void speedup(int increment) {
        speed += (increment*2);
        System.out.println("Speeding x2");

    }
}

public class Main {

    public static void main(String args[]){
        SpecialBike A = new SpecialBike();
        A.setGear(5, "up");
        int brake = A.applyBrake(2);
        A.speedup(2);
    }
}
```

Figure 1

Continue...

a) From the code in Figure 1, identify TWO **compile** errors (state the line number in your answer) and explain why it is considered as error. [2 marks]

b) Explain what *object* is in the context of Object-Oriented Programming. Based on the code in Figure 1, give one example of *object*. [2 marks]

c) Supposed that the code does not contain any error, what is the output of the code when executing line 38. Explain your answer. [2 marks]

d) Write one default constructor for class `Bicycle` and one default constructor for class `SpecialBike`. The constructor for class `Bicycle` should set the private members to zero. [4 marks]

Question 2 (10 marks)

a) Explain the concept of encapsulation or "information hiding" in object-oriented programming. [1 mark]

b) Write an interface `printable` that will have one `void` method `print()`. Then write a class `A4` that implements the `printable` interface. The method `print()` will display `Hello` as output. [4 marks]

c) Write a user-defined exception class `MyException` that extends the `Exception` class. Define two constructors (default and parameterized with `String` parameter) that will call the constructor of the base class using the word `super`. The message that will be passed for default constructor is `"Exception has occurred!"` [3 marks]

d) Write the output of the following code in Figure 2 [2 marks]

```
public class Donut
{
    public static void main(String [] args)
    {
        int a = 5;

        try
        {
            countDonut();
            System.out.print("A");
        }
```

Continue...

```
        catch (RuntimeException ex)
        {
            System.out.print("B");
        }
        catch (Exception ex1)
        {
            System.out.print("C");
        }
        finally
        {
            System.out.print("D");

            if(a == 3){
                System.out.print("E");
            }
        }
        System.out.print("F");
    }

    public static void countDonut() throws Exception
    {
        throw new Exception();
    }
}
```

Figure 2: Exam class

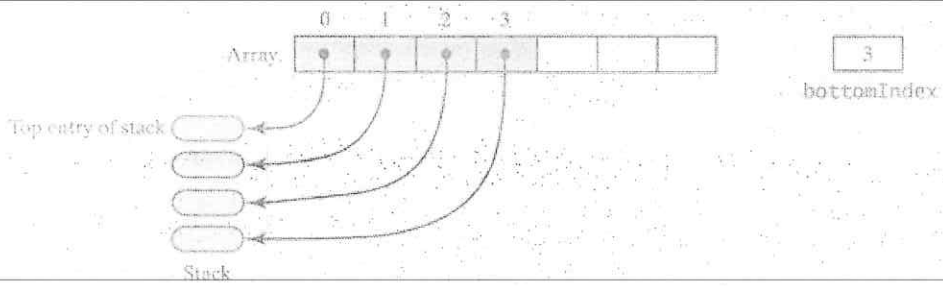
Question 3 (10 marks)

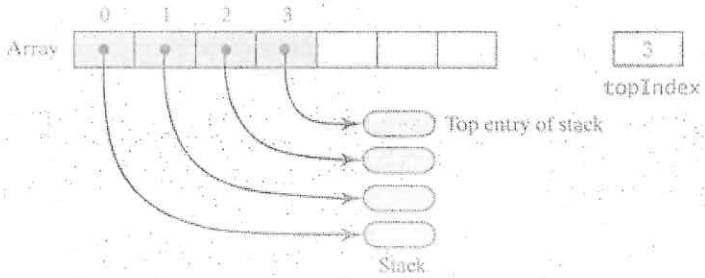
a) Given the following AList class, write the *addtoback* method for array implementation of a list. [3 marks]

```
public class AList < T > implements ListInterface < T >,
java.io.Serializable
{
    private T [] list; // array of list entries
    private int length; // current number of entries in list
    private static final int MAX_SIZE = 50; // max length of
                                            // list

    public boolean addtoback (T newEntry)
    {
        // write the implementation of this addtoback method

    } // end addtoback
}
```

Continue...

b) The following figure shows two methods of implementing a **stack** using array. The first method uses index 0 as the top entry of the **stack** where as the second method uses index 0 as the bottom entry of the **stack**. In your opinion, which method is better. Give a reason for your answer. [2 marks]





c) Initially, `myPriorityQueue` is empty. Write the content of the priority queue assuming that the priority is interpreted as coming first in a lexicographical ordering. [2 marks]

```
myPriorityQueue.add("Jim");
String name = myPriorityQueue.peek();
myPriorityQueue.add("Jane");
myPriorityQueue.remove();
String name = myPriorityQueue.remove();
myPriorityQueue.add("Jill");
myPriorityQueue.add("Jim");
myPriorityQueue.add(name);
myPriorityQueue.add("Jess");
```

d) Using a diagram, briefly describe the doubly linked implementation of a Deque when adding a new node to the Deque that has some existing elements. [3 marks]

Continue...

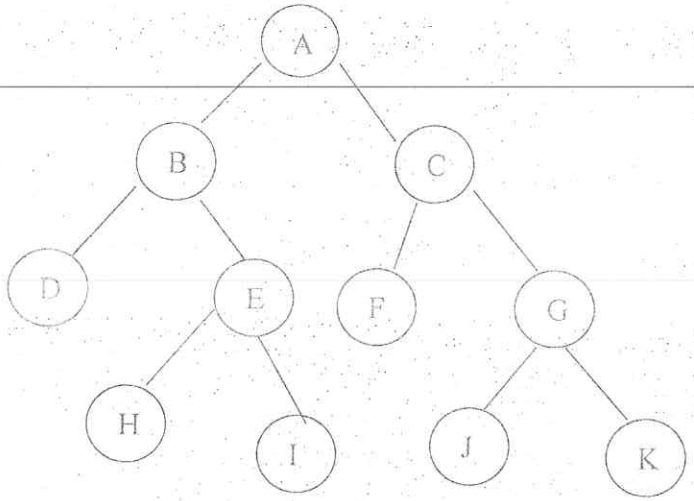
Question 4 (10 marks)

a) Given the following array, describe using diagram how binary search can be performed if number 92 is to be searched. Assume that the most left entry is index 0. [3 marks]

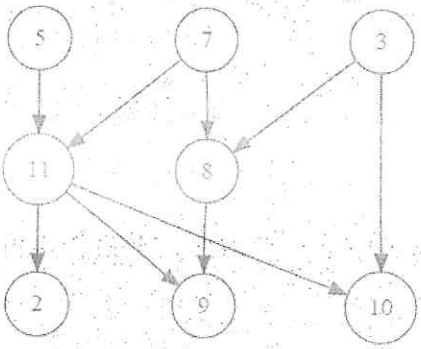
16	23	33	41	57	66	69	74	76	77	78	92
[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	[9]	[10]	[11]

b) Write the algorithm for selection sort. [2 marks]

c) Given the following tree, list the visitation order for preorder, inorder and post traversal [3 marks]



d) Given the following directed graph, find two topological orders. [2 marks]



End of Page.